PROGRAMMING FOR A DIGITAL COMPUTER CONTROL SYSTEM

C. H. GOULD

1953

PROGRAMMING FOR A DIGITAL COMPUTER CONTROL SYSTEM

by

Charles Henry Gould
Captain, United States Marine Corps

Submitted in partial fulfillment
of the requirements
for the degree of
MASTER OF SCIENCE

United States Naval Postgraduate School
Monterey, California
1953

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

from the

United States Naval Postgraduate School

PREFACE

The work described herein was done at the Research and Development Laboratories, Hughes Aircraft Company, Culver City, California, during the industrial term of the Electronics Engineering curriculum. My control system project was suggested by the Laboratories, and was carried out with their fullest cooperation as a part of a large scale study of airborne digital control systems.

I shall assume that the reader is conversant with current terminology in the computer field (2). What few terms are not in general use I will explain. The words "order" and "instruction" are synonymous.

TABLE OF CONTENTS

Preface	ii
Introduction	1
Chapter I. Description of Equipment	3
Chapter II. The Problem and Its Programming.	
1. The Problem	5
2. The Equations	7
3. The Fly Program	10
4. The Entry Program	14
5. The Test Program	15
Chapter III. Coding The Problem	17
Chapter IV. General Discussion and Recommendations	21
Bibliography	39

LIST OF ILLUSTRATIONS AND TABLES

Figure I. Schematic of Control Box.	Envelope
Figure II. Geometry of Fly Problem.	26
Table I. Coding Sheet for Digital Computer Problem	Envelope
Table II. Instructions Available in Model I Digital Computer	27
Table III. Flow Diagram of Fly Instructions	28
Table IV. Program of Fly Instructions	29
Table V. Program of Entry Instructions	37

I. INTRODUCTION

This thesis describes in detail a program and coding for a particular aircraft navigation display control system utilizing a digital computer, in less detail the equipment used, and finally a number of suggestions for programming of control systems.

Some modern control systems become very complex and require many individual computing or compensating elements. It seems likely that a gain might be achieved by combining all these computing and compensating functions in one elaborate general-purpose computer, perhaps on a time sharing basis (1). Either digital or analog techniques could be used.

A digital computer offers certain inherent advantages in this application. Chief among these are, first, that changes in function of parts of the control system, such as adjusting compensation of a servo loop, are made simply as changes of numbers or instructions within the computer. No change of physical equipment is needed. Second, and more basic, is the memory, or storage of information that a digital computer contains. Thus, the control system can be influenced by its past performance, or by outside stimulation. Take, for example, an automatic factory. Then the output of the factory could be influenced by inventories and sales records which had been gathered and collated by itself.

Such an elaborate control problem is presented in some high-performance aircraft, and the computer on which the described program was

run was a digital computer for airborne use (3).

Programming and coding a problem for a digital computer have received much attention (4), but the approach has been one of reducing to a minimum the human labor, either by admittedly inefficient computations (5,pg.115) or by making the machine do the work of coding (6). This approach leads to overall efficiency when a problem is done once or a few times. If, however, a problem is repeated ad infinitum, as in a control system, human planning time starts to pay off as reduced physical equipment due to the lessened storage capacity needed for the more efficient set of instructions. This will be discussed later.

In any machine that is very complicated some means of diagnostic testing should be provided. This will be elaborated..

Finally, a reasonably didactic description of the program will be given. This program was a first attempt, and it is on the basis of the mistakes made that the recommendations have been formulated.

CHAPTER I. DESCRIPTION OF EQUIPMENT

The equipment used, while not the subject of this thesis, did influence the programming and coding, and so should be described. Peculiarities of the computer logic important to the program will be described in later chapters when applicable.

The computer used was a first model which had been used for some time for circuit and component reliability and logical checks. No input-output equipment had been designed specifically for it, although input-output equipment was available that had been used on a very similar computer. One function of the Control Box (Chap. II) was to adapt and interconnect the computer and the input-output equipment. The Brown Teleplotter was used intact except for changes in its internal servo system to respond to dc signals of the desired amplitude.

The computer is of the general purpose, serial, binary digital type (3). It has an arithmetic unit, a control unit, and a magnetic drum memory unit. The arithmetic unit performs the arithmetic operations of addition, subtraction, multiplcation and division. The unit consists of three one work circulating registers and a binary adder. The magnetic drum memory (8) provides one eight word circulating register, the "d" register, and fourteen sixty four word storage bands. These bands are interpreted by the control unit as eight order bands containing instructions, and six number bands. One of the number bands and one of the order bands are tied together, so that an instruction can be interpreted as a number and changed, written back into the memory, and then used as a new instruction. Such changes are made as

the result of the computations in progress. The instructions available to the coder are shown in Table II.

The input-output equipment converts analog information (dc voltages) into an equivalent binary number at a sampling rate of once every fourteen magnetic drum revolutions, writes the most recently sampled binary number onto the drum, reads the most recently written output binary number (written by the computer) and converts it to an equivalent analog quantity (dc voltage). It is multichannel, having nine inputs and four outputs, with one calibrating period (7).

The plotting board consists of a chart and pen so arranged that two input dc voltages are interpreted and plotted as x and y deflections of the pen on the chart. A potentiometer error detector feeds a modulator, whose ac output is proportional to the error. An amplifier then drives a motor in such a direction as to tend to make the error zero. This is duplicated for the two axes. The speed of response of the pen is much faster than the contemplated plotting speed.

CHAPTER II. THE PROBLEM AND ITS PROGRAMMING.

1. The Problem

A navigation display was desired to demonstrate the ability of the Hughes Model I digital computer to store much information for a long time. The operation of the device was visualized as follows. First, a map of the desired track is placed on a plotting board containing a movable plotting pen. The pen is placed in turn over the departure point and each of the enroute way points, or "fixes". At each point a button is depressed which will enter the position into the memory of the computer. After all points are entered, the pen is returned to the departure point and the computations started by throwing a switch. The pen then "flys" to each of the fixes in turn at a fixed ground speed on straight line courses. After getting to the last fix, or destination, the computer would turn itself off after preparing the program for entering the next set of fixes. Provision was made for entering sixteen fix positions. A signal light would indicate when the computer was ready to have fixes entered, or ready to fly.

The Control Box had the following operational controls: two handwheels for East-West and North-South positioning of the pen during the entry portion of operation, a pushbutton for entering fixes and starting fly operation when ready, a function switch with Enter and Fly positions, and a switch to turn on recording pen on plotting board to

prevent smearing. Contained in the same box were various interconnections between the computer, the input-output equipment, and the plotting board, various disconnect switches for the power and timing pulses, and reading and writing amplifiers required by the input-output equipment. A schematic of this box is included as Figure 1 to indicate the functions, although the details are immaterial to this thesis.

It was desired to make the fly routine a feedback loop so that the present position of the pen could be sensed in the computations, and so that smoothing could be included by placing a low pass filter after the dc analog output. Smoothing could have been done in the program, and probably will be included at a later date by Hughes engineers. Smoothing in the program would insure, for example, that computational errors of a random nature would be smoothed out of the output, rather than being cumulative. Smoothing in the dc analog output will accomplish the same result, but requires additional equipment. Smoothing should not be confused with the essential filter required to smooth the sampling rate.

In the course of the fly computations, the present position is written into the memory by the input equipment. The computer then reads this position and compares it with the fix position to which the "plane" is now headed. By a trigonometric method involving the fixed speed towards the fix position, an incremental change in position for

the computing period is determined and added to the present position to give a future position. This future position is then fed to the output equipment, through the low pass filter, and to the pen to position it. This future position then becomes the present position for the next computation. The computations are repeated every 14 drum revolutions or 0.1 seconds, i.e., the sampling rate is ten cps. Note that if a computational error is made and the plane gets off course, it will still continue to fly to the fix on the nearest straight line course. When the plane arrives at the desired fix, the computer will change its instructions so that the next fix position will be used. After getting to the sixteenth fix position the computer will stop, after setting in the proper instructions to enter new fixes.

2. The Equations

The course and incremental position are computed by the following set of equations.

Let x, y be the coordinates of the present position.

x_p, y_p be the coordinates of the next fix position.

dt be the sampling period, 0.1 second.

dx, dy be the coordinate distances travelled in time dt.

V be the specified ground speed towards the next fix.

θ be the angle of direction of travel, measured from the x axis.

Then we can say that

$$\frac{Y_p - Y}{x_p - x} = \tan\theta = dy/dx.$$

$\cos\theta = 1/\sqrt{1 + \tan^2\theta}$.

$dx = Vdt \cos\theta$.

$dy = dx \tan\theta$.

$x_{n+1} = x_n + dx$.

$Y_{n+1} = Y_n + dy$. These last two are the future position, and become x, y for the next period.

Difficulties arise from two sources. (1) Ambiguities due to sign of $\tan\theta$, and (2), the fact that the computer can handle only numbers whose magnitude is less than one. These will be taken care of by using the conditional transfer ckS instruction. (Table II). Note that no instruction is provided to specifically handle square roots, so an approximation will have to be made for $\cos\theta$.

The equations finally used that follow, resolve these difficulties. See fig. 2.

Given x, y, x_p, Vdt, Z.

1. Find $|X_p - X|$.
2. Find $|Y_p - Y|$.
3. Find $|X_p-X| + |Y_p-Y|$. Store for future use.
4. Find $|X_p-X| - |Y_p-Y|$. Check sign. If

5a. Positive. Find

$(Y_p-Y)/(X_p-X) = \tan\theta$.

6a. Approximate $\cos\theta$ by polynomial.

7a. Multiply $Vdt\cos\theta = dx$.

5b. Negative. Find

$(X_p-X)/(Y_p-Y) = \cot\theta$.

6b. Approximate $\cos\theta$ by polynomial.

7b. Multiply $Vdt\cos\theta = dx$.

8a. Multiply $dx \tan\theta = dy$.

8b. Divide $dx/\cot\theta = dy$.

9. Divide $dx/(X_p - X)$. Check sign. If

10a. Positive. Do nothing. $dx = dx^*$.

10b. Negative. Subtract $0 - dx = dx^*$.

11. Divide $dy/(Y_p - Y)$. Check sign. If

12a. Positive. Do nothing $dy = dy^*$.

12b. Negative. Subtract $0 - dy = dy^*$.

13. Add $X + dx^* = X_{n+1}$. Write as output.

14. Add $Y + dy^* = Y_{n+1}$. Write as output.

15. Subtract $(|X_p - X| + |Y_p - Y|) - z$. Check sign. If

16a. Positive. Start over with same X_p, Y_p.

16b. Negative. Start over with new $X_p + 1$, $Y_p + 1$.

17. Count up a number such that after executing 16b sixteen times, a change of sign occurs to stop computer.

In the above, step 4 determines whether $\tan\theta$ is going to be greater than one. If so, $\cot\theta$ is used. Steps 9 and 11 insure that dx and dy are of the proper sign. Step 15 checks how closely the plane is to the fix position. When this distance, or rather a distance which is slightly greater than the distance to the fix, becomes less than some small number z, the plane has arrived, and the next fix position is used.

The polynomial used to approximate $\cos\theta$ from $\tan\theta$ or $\cot\theta$ is derived by a Chebyshev polynomial operation, and is accurate to 0.3%

within the range used:

$$\cos\theta = 0.1474x^4 - 0.4340x^2 + 0.9965, \text{ where } x = \tan\theta, \text{ or}$$

$$\cos\theta = y(.1474y^4 - .4340y^2 + 0.9965), \text{ where } y = \cot\theta.$$

This approximation is easily done on the computer, and is easier than conventional algebraic methods would be.

3. The Fly Program

Using the instructions contained in Table II and these equations for a guide, the program was designed and is presented in Table IV. Table III shows a flow diagram of the instruction sequence. Before detailing the instructions some general explanations are required.

On a machine which has available only such unsophisticated instructions as shown in Table II, the detailed coding must proceed closely with the program, since the order in which operations are performed depends upon, for example, when certain numbers become available from the memory. It may be advantageous to delay some operations till a more convenient time. Thus the sequence of instructions may seem odd until the coding is studied in Chapter III.

The subscript on instructions involving the D register denotes the sector number in this eight word memory.

The B register acts as an accumulator for the arithmetic unit; that is, any number written into the B register adds algebraically to the number already there. In starting a new operation it is necessary to insure that the B register is cleared, as in instruction # 1.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Any instruction involving "from B", such as ba, bm, etc., will clear the B register.

All other registers and the memory will hold a number until a new number is written over it, replacing the old number with the new number.

Instruction #1 will be called simply #1, etc.

#1 clears the B register.

#2 transfers the fix X_p to the D register. This instruction will have to be changed as we progress from fix to fix. # 3 does same for Y_p, but will be a series of instructions, as will be seen in Chapter III.

#7 and #8 bring in the uncorrected values of present position, x_u and y_u, from the memory. These numbers are then divided by a scale factor (7) to become the corrected x and y, and are stored in D (up to #17).

#18 to #38 are manipulations to obtain $K'+L'$, which is a rough measure of the distance to go to the fix, see Figure 2. Some intermediate results are stored in D for future use.

The quantity $K'+ L'$ can be a number from 0 to $+2$. This computer, which is designed to handle numbers between -1 and $+1$ (modulo 2), interprets a number between 1 and 2 as a negative number. Thus #39 checks this fact, and if negative, indicating that $K'+L'$ is greater than 1, writes in its place an arbitrary positive number, $\frac{1}{4}$.

#44 to #49 form $K'-L'$ as a measure of the octant of θ. If in an

octant such that tan θ is less than 1, the computations use tan θ. If tan θ is greater than 1, the computations use cot θ. This avoids overflow on the following division. #50 senses this fact.

#51 to #54 form tan θ, OR #73 to #76 form cot θ.

Whichever form is used, #55 to #72 form either cos θ or cos θ/cot θ by the approximation shown on page 10.

#78 and #79 bring K'-L' into the A register again to check sign. This was an easier operation to code than to store the actual decision, #50, previously made, i.e., the decision was made twice.

#81 to #93 form dx and dy for the case when cot θ is used. The sign of dx and dy may be wrong.

#94 to #102 form dy and dx for the case when tan θ is used. Again, the sign may be wrong.

#103 to #118 check that dx is in the same direction (has same sign) as x coordinate distance to the fix. If wrong, the sign is changed by subtracting from zero.

#119 to #137 does same for dy. Note again that a definite possibility for overflow exists in the two divisions, which are used to check parity of sign, when the present position approaches a fix. dx and dy must be larger than the actual change in present position due to the time lag effect of the smoothing filter. This mistake was corrected by utilizing a multiplication to check parity of sign in the actual machine, but was left in this program to illustrate that not only must

you think in the language of the computer, but also in terms of the overall machine considered as a control system, i.e., its analog aspects.

#138 to #149 form the future position and write this position into the memory in a position where the output conversion device can read it.

#150 to #154 compare K'+L', or roughly distance to fix, with a small arbitrary number, z. When the pen has approached within z of the fix position #155 senses this change of sign. If not within z of fix, several waits are necessary so that recycling back to the #1 start of the program will be after the 14th revolution of the magnetic drum. This is to insure synchronism with the sampling rate of the input-output equipment.

If within z of fix, #160 to #165 take #2, add a number to the instruction such that the "words to wait" (see Chapter III) are changed to give a new X_p and Y_p, and write the new #2 back in position so that it can be used the next time around. Each time a fix is changed a number is added to the count number such that after the sixteenth fix is passed the count number becomes greater than 1, overflows, and appears as a negative number to #175. #179 then writes 0 in output 4, which is connected through the output device to a signal lamp, putting out the lamp. Later we will see that the lamp had been put on after the

last fix was entered. #180 to #182 write in #2 of Table V, the entry program (see next section), so that the instructions will be correct for its initiation.

#183 uses the same order as some number not zero so that #184 will stop the computer after getting to the sixteenth fix, or the destination.

4. The Entry Program

In Table V are the instructions used to enter fixes into the memory.

#2 to #6 change the "words to wait" in the same manner as in previous section. Since this operation is done before using the order, the first "old am instruction" will be some number, meaningless as an instruction, which when added to will be the desired "new am instruction". This will be discussed further in Chapter III.

#7 to #16 alter present position, which in this case is the desired fix to be entered, by the same scale change as in previous section.

#17 is an example of programming influenced by the coding requirements. It was necessary to enter this number into the same position in the memory as the "old am instruction" had been. This required sixty four word times, and so the intervening time might as well be used for some more useful purpose.

#21 makes use of the number written by #17 as an instruction, and enters the fix position into the memory.

Since the instruction following #22 must occur immediately after writing the fix position (See Chapter III), and the fix positions are in separate places in the memory, #23 becomes a series of identical instructions, but with different "words to wait". This provides a means of determining when the sixteenth and last fix has been entered without utilizing a count number as was done in the previous section. The last of this series is called #28.

For the first fifteen points, then, #23 to #27 puts an arbitrary positive number in to B, a zero into output 4 for the signal light, and stops the computer ready for the next fix to be entered.

#28 to #35 writes the instruction which becomes #2 of the fly program in its proper place, writes a number which is nearly full scale into output 4 to light signal light, writes a zero in the count number used in the fly program, and stops the computer ready for flying.

5. The Test Program

The test program shown in Table I was included for diagnostic testing of the computer components. It was of a type used previously for testing the Model I computer, and performs the arithmetic functions, various transfers, and utilizes all the number and order bands of the memory. It does these operations on all binary numbers from 2^{-7} to $1-2^{-7}$.

This program is not explained in detail since it is not the writers work, and is immaterial to this control system, except in its operational use.

CHAPTER III. CODING THE PROBLEM

Coding the program developed in Chapter II requires a detailed knowledge of the computer and how instructions follow one another. Table I, contained in the envelope in the back of the thesis, lists the coded problem. Programming and coding were done nearly simultaneously.

Three separate problems are coded in Table I: the fly program whose instructions are labelled F, the entry program labelled E, and the test program labelled T. The instructions in each of the three are serially numbered whenever possible. Table III shows a flow diagram of instructions for the fly program. Reference should be made to Tables IV and V for the sense behind the coding.

The Model I computer uses a relative address code (3) in which the address of the next instruction is given by specifying the memory band in which the instruction occurs, and the number of word times till the new instruction is ready to be read. The system is relative because the sector address of an instruction (or number) is never given, except for the first of a series of instructions, which always is in sector 1 in the selected memory band. (The m^{th} sector is the m^{th} word after the arbitrarily designated sector 64 in memory).

An example will follow to illustrate this point. First, the "bookkeeping" shorthand should be explained. The instruction itself is follow-

ed by a series of three numbers in Table I. The first is the words to wait after reading the order, and before executing the order. The second number specifies the number band of the memory involved if meaningful; thus, the instruction $\neq$ does not involve the memory, but am does. The third number specifies the memory band of the next instruction. For example, in sector 06, band 1, the instruction md_1 2,4,1 3T occurs. This means that the number in number band 4 should be transferred to the D register, sector 1, after a 2 word wait, and that the next instruction, #4T, will be found immediately after execution in band 1. The instruction was then read out during sector (word) #06, the computer did nothing during sectors #07 and #08, transferred the number found in number band 4, sector #09, to the D register during sector #09, and read instruction #4T during sector #10. Note that the D register is eight words long, and arbitrarily sector 1 of the D register coincides with sector 1 of the sixty four word memory. Thus the D register repeats itself starting with sectors 9, 17, etc. of the memory.

The permissible words to wait before execution contained in any instruction can be anything from 0 to 16 words, except the wait instruction (wt) which can have 0 to 32 words wait. The wait instruction is executed immediately, or rather it would be better to say that execution corresponds to reading the next instruction. For example,

check instruction #159F in sector 52, band 2, which feeds into instruction #1F.

Note that a particular operation to be performed by an instruction may be wrong. As long as the address of the next instruction is correct the computer will continue to cycle instructions blindly. This is of particular importance in trouble shooting. The usual procedure is to make the instructions cycle, then worry about what mathematical results are obtained.

The three programs all start in sector #1, but in different memory bands. The starting position is selected either by a set of buttons on the computer, or by the last instruction read by the computer. Thus, when instruction #148F is read and the computer stops ready to start on the entry program, the address of the next instruction is band 3. When the computer is started again by pushbutton, the instruction #1E will be read first.

It is by means of the variable words to wait portion of an instruction that sixteen different fix positions are entered and taken out of the memory. However, in the actual number representing a particular instruction, the compliment of the words to wait appears. Thus, the four digits representing the words to wait will appear oddly when added to in the manner shown. When all four digits are zero, this means 16 words to wait; adding one least significant digit means 8

words to wait, etc. See number band 2.

Rather than taking up the detailed coding step by step, the interested reader may follow through as much of Table I as necessary to get the system in mind. Compare Tables I, III and IV with emphasis on the operation of the conditional transfer instruction ckS and the variable words to wait.

It must be obvious to anyone who has followed the coding in any detail that there is much wasted time and space here. The problem has become not one of electronics or control system design, but rather one for an ingenious bookkeeper and for numerical analysis. Time did not permit completion. If this control system were to be used in an important application, however, such effort would be essential to a good design.

CHAPTER IV. GENERAL DISCUSSION AND RECOMMENDATIONS

The test program used in this device was a good one in that all functions of the computer were tested. However, diagnostic indication of troubles encountered was very limited. A test should give a positive indication to a serviceman of the location of troubles. The instructions available in even as unsophisticated a computer as the Model I should allow this by making more liberal use of the ck0 instruction throughout the test program shown in Table I. Since a digital computer is so literal and human coding so liable to error, one must know whether the machine is at fault or not before spending the hours required to trouble shoot a program for a control system.

Once assured of proper machine operation, several steps can be taken to get a program to cycle instructions correctly. As noted previously, it is usual to cycle instructions, then run test problems to check mathematical operations. After entering a set of instructions into the computer in which a mistake is made, improper cycling will usually result in circulation through an improper sequence of instructions. If this occurs, the following steps are of value with the Model I computer.

1. Change an instruction, say about half way through the program, to a ck0 at some point where the computer will surely stop. If this half of the instructions cycles correctly and gives proper intermediate results from a test problem, put the instruction back and change another instruction say three quarters of the way through. Continue this pro-

cedure till the trouble is narrowed down.

2. After narrowing down, change one or more instructions to cycle only a small portion of the program repeatedly, and get it to work. This might be called reentrant checking.

3. Another method after narrowing down is to use test equipment built into computer (provided in most computers) which will allow execution of a single instruction at a time. Check each step in a suspected region.

4. In troublesome cases it may be necessary to check each instruction with an oscilloscope. As might be suspected, this is time consuming and should be avoided if possible.

5. A good deal of ingenuity helps more than set rules.

When proper cycling is accomplished some previously prepared test problems should be used to check operations. The method used with this set of programs was to disconnect the input equipment, write in known numbers where the input numbers would have been written, and check either the final output, or the numbers held in the registers at some intermediate time, using the narrowing down technique previously described.

If now the computations are properly made by the computer, the rest of the control system comes under scrutiny, especially the stability of the system and errors. Note that the program in use is an integral part of the control system, and must be thought of as part of the equipment.

A useful concept in present control systems is the transfer function

either of the system or its components. A program may be thought of in terms of a transfer function involving principally functions of the term exp(-as), a time lag (9). Unfortunately, too little has been done to develop a theory for this concept. It is mostly a study of numerical analysis (9).

Damping of a control system has an analog in a digital computer program. Derivatives and integrals are replaced in these discrete calculations with differences and sums. Damping then becomes smoothing, which may take the form of averaging or interpolation, or approximation of the required derivatives by differences. The problem is one of numerical analysis and will be discussed further only in connection with noise.

A simple method of estimating stability was suggested by Dr. Jacobi for the control system described. Since it is a feedback system involving positive feedback, oscillation could occur when the total phase shift became a multiple of 2pi. The fly program has a time delay of 0.1 second, or a phase shift of $\phi_1 = W/10$, where w is the angular frequency. The RC filter used on the output equipment has a phase shift $\phi_2 = wRC$, where RC = 0.5. Oscillation at w = 0 is meaningless. $\phi = \phi_1 + \phi_2 = 2pi$ occurs when w = 15 pi, or at a frequency of 7.5 cps. However, the RC filter has an attenuation of 27 db.at this frequency so no trouble was expected or encountered. When the filter was disconnected for trial, oscillation, or at least random action did occur although the pen would of course not follow the gyrations of the numbers in the computer. Note that the same

smoothing effect could have been attained within the program had it been designed into the program.

Errors within the computations (excluding a steady bias error) may arise from two principal sources. Round-off errors appear as white noise. Large random errors may occur due to electrical interference or errors in programming (see pg. 12), or due to marginally satisfactory equipment within the computer. The random errors may be reduced by proper design and maintenance, but probably cannot be eliminated. Both types of noise can be reduced within the program by smoothing. White noise is most effectively reduced by some means of averaging or interpolation. Large random errors can be treated, like ignition noise, by limiting the value of successive increments in an iterative process, e.g., limiting dx and dy in the fly problem.

An error which occurs as a bias error, e.g., a wrong scale factor, can cause considerable trouble in a system involving feedback. Scale factor is contained in the input and output equipments, within the program, and in any analog components external to these, particularly the sensors. Thus the accuracy of the overall system is still determined by the weakest link. To take advantage of the possible accuracy of most digital computers, then, attention should be directed to the design of the analog components of the system, as well as to the program. In a time-sharing system it would be of value to utilize one computing

cycle for recomputing and writing a new scale factor into the program periodically, i.e., automatic calibration.

Using a digital machine implies both quantizing and sampling; these should be consistent with system design. When high sampling rates are required, e.g., high roll rates of present jet aircraft, a stringent limitation is placed on the program designer and the computer engineer to make an efficient (few words per cycle) and rapid (many words per second) machine. Low quantizing error may be required in some applications. These two conditions are unfortunately contradictory in a serial machine. This is especially unfortunate when one equipment is used on a time-sharing basis for many jobs. The use of a parallel machine may be required with its higher speed of operation but greater amount of equipment, if the program designer is not dexterous enough.

A program may be made more efficient by efforts of the numerical analyst in reducing the steps required to do the series of operations. The coding of a particular program is a related but different problem. In coding, each wait for the appearance of a number in the memory is inefficient, as are completely filled sectors when other sectors are relatively vacant (compare sectors 5 and 55 of Table I). The requirements of coding will probably determine the details of the computer design in the long run. Efficient coding is the result of attention paid to a great many small details.

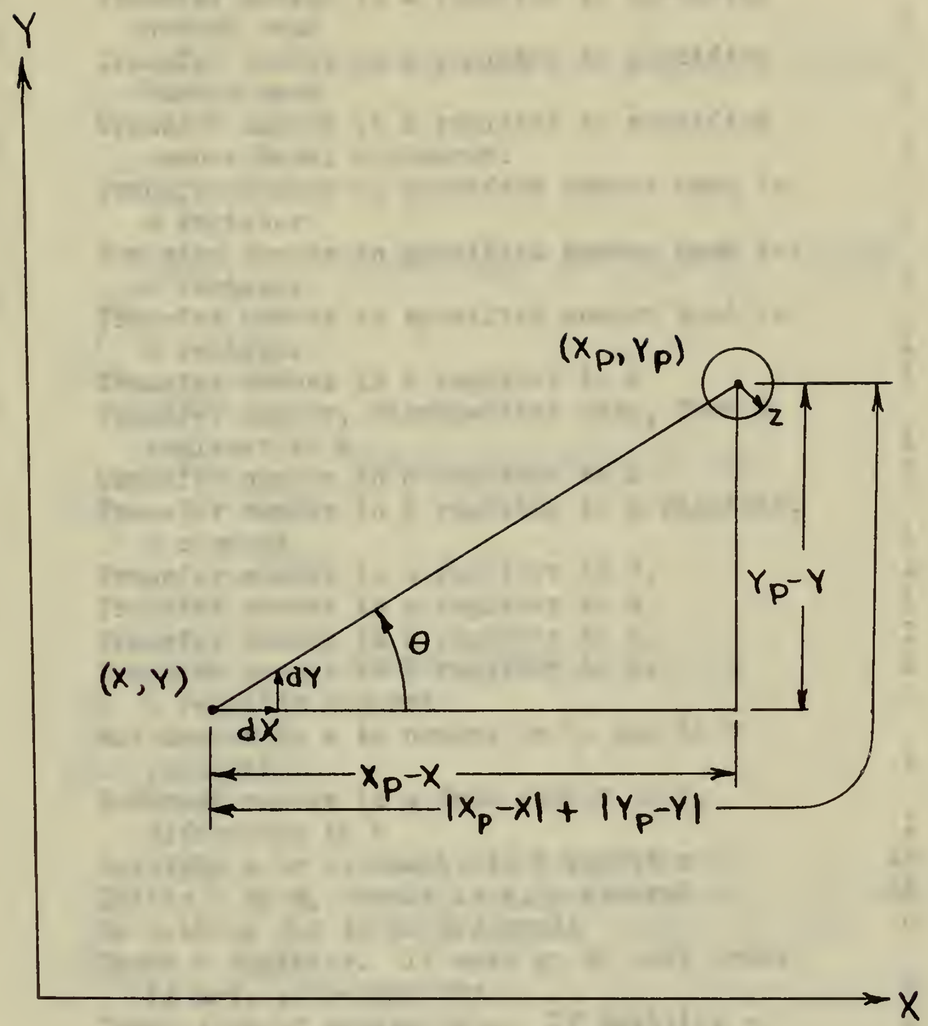


FIGURE II. GEOMETRY OF FLY PROBLEM.

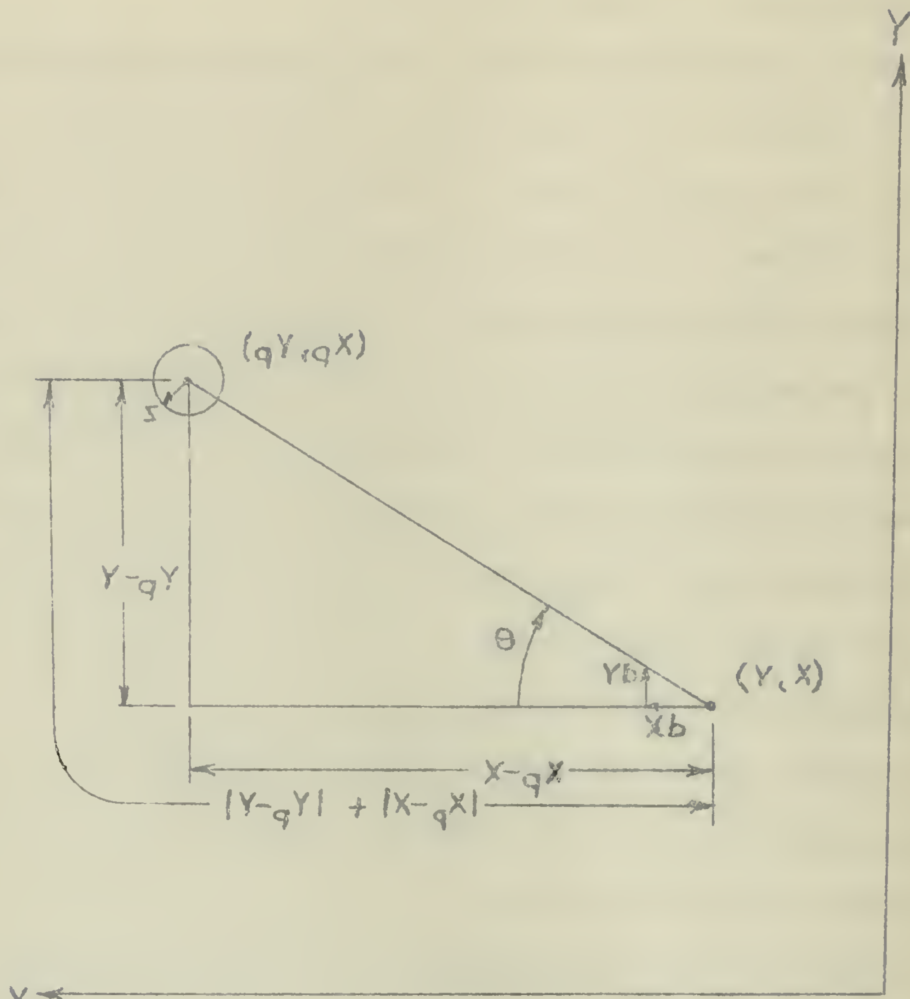


FIGURE 11. GEOMETRY OF FLY PROBLEM.

Symbol	Meaning	Words to Execute
cm	Transfer number in c register to specified memory band	1
am	Transfer number in a register to specified number band	1
dm.	Transfer number in d register to specified number band	1
bm	Transfer number in b register to specified number band. b cleared.	1
ma	Transfer number in specified number band to a register	1
mc	Transfer number in specified number band to c register	1
md	Transfer number in specified number band to d register	1
ca	Transfer number in c register to a	1
Aca	Transfer number, disregarding sign, from c register to a	1
da	Transfer number in d register to a	1
ba	Transfer number in b register to a register. b cleared	1
cd	Transfer number in c register to d.	1
ad	Transfer number in a register to d.	1
dc	Transfer number in d register to c.	1
bd	Transfer number in b register to d. b register cleared	1
+	Add number in a to number in b, sum in b register	1
-	Subtract number in a from number in b, difference in b	1
X	Multiply a by c, result in b register	16
div	Divide b by a, result in c, b cleared	16
wt	Do nothing for (1 to 32) words	0
ck0	Check b register. If zero go to next order. If not, stop computer	1
ckS	Check sign of number in a. If positive - -	1
	If negative - -	2

Table II. Instructions available in Model I Digital Computer.

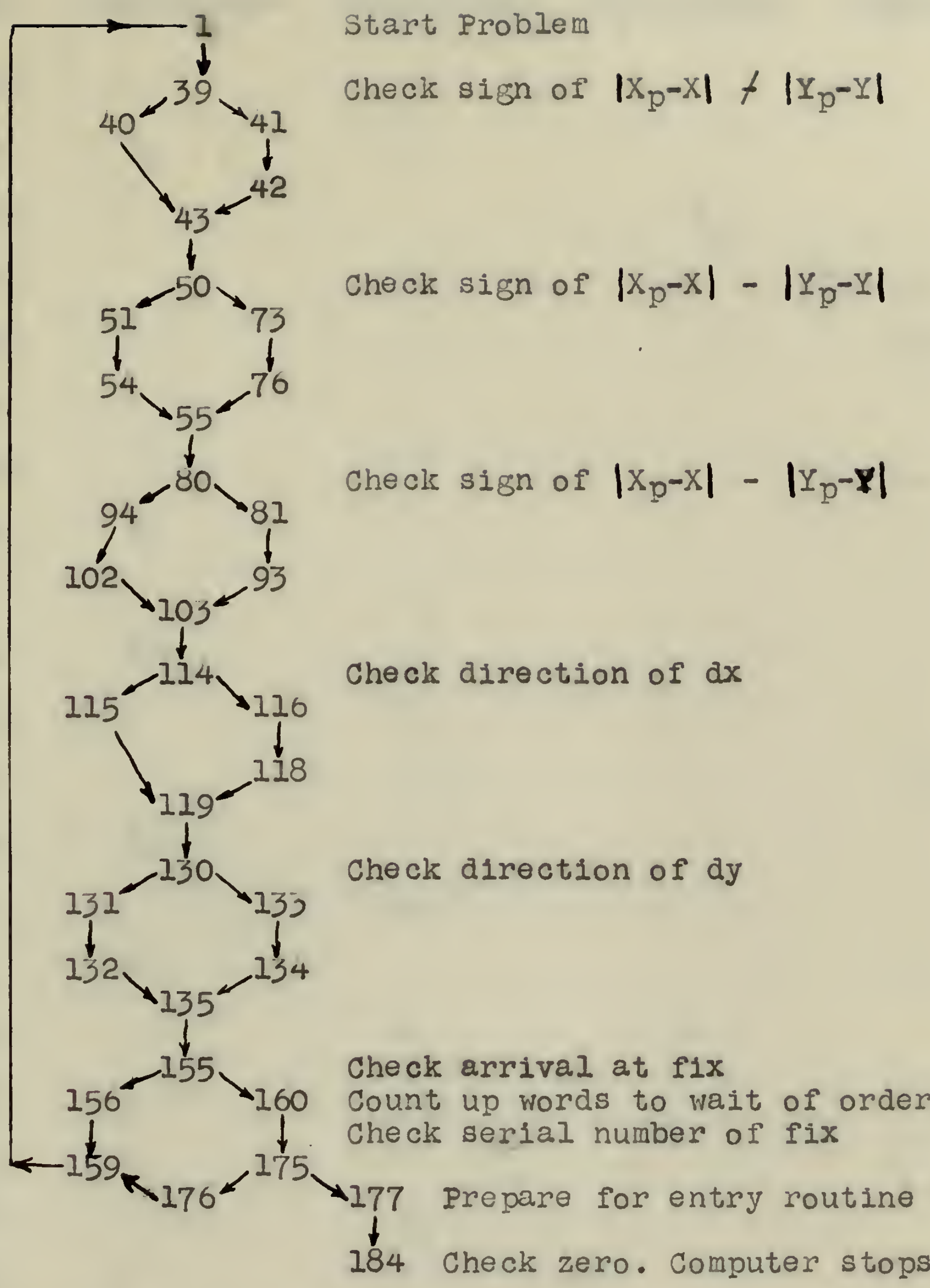


Table III. Flow Diagram of Fly Instructions

1 Start Problem

39 Check sign of X_p-X + Y_p-Y

40 41

42 43

50 Check sign of X_p-X - Y_p-Y

51 73

54 76

55

80 Check sign of X_p-X - Y_p-Y

94 81

102 93

103

114 Check direction of dx

115 116

118

119

130 Check direction of dy

131 133

132 134

135

155 Check arrival at fix

156 160 Count up words to wait of order

Check serial number of fix

159 175

176

177 Prepare for entry routine

184 Check zero. Computer stops

Table III. Flow Diagram of Fly Instructions

Inst.	No. in A Reg.	No.in B Reg.	No.in C Reg.	No.in memory
1.ba		0		
2.ma	X_p			
3.mc			Y_p	
4.cd_8				Y_p in D_8
5.ad_3				X_p in D_3
6.wt				
7.md_6				X_u in D_6
8.ma	Y_u			
9.$\div$		Y_u		
10.ma	S.C.			
11.div		0	Y	
12.cd_7				Y in D_7
13.d_6a	X_u			
14.$\div$		X_u		
15.ma	S.C.			
16.div		0	X	
17.cd_6				X in D_6
18.d_3a	X_p			
19.$\div$		X_p		
20.d_6a	X			
21.-		$X_p - X = K$		
22.bd_2		0		K in D_2
23.d_2c			K	
24.Aca	$K' = \lvert K \rvert$			
25.ad_1				K' in D_1

Table IV. Program of Fly Instructions.

Inst.	No. in A Reg.	No. in B Reg.	No. in C Reg.	No. in memory
26. d_8a	Y_p			
27. +		Y_p		
28. d_7a	Y			
29. -		$Y_p - Y = L$		
30. bd4		0		L in D_4
31. d_4c			L	
32. Aca	$L' = \lvert L \rvert$			
33. ad_5				L' in D_5
34. d_1a	K'			
35. +		K'		
36. d_5a	L'			
37. +		$K'+L'$		
38. ba	$K'+L'$	0		
39. ckS	If +, to 40 If -, to 41			
40. wt				
41. ma	$\frac{1}{4}$			
42. wt				
43. am				$K'+L'$ or $\frac{1}{4}$ in m.
44. d_1a	K'			
45. +		K'		
46. d_5a	L'			
47. -		$K'-L'$		
48. ba	$K'-L'$	0		
49. am				$K'-L'$ in m.
50. ckS	If +, to 51 If -, to 73			

Table IV. Continued

Inst.	No.in A Reg.	No.in B Reg.	No.in C Reg.	No. in memory
51.Aca	L'			
52.+		L'		
53.d_2a	K			
54.div		0	$L'/K = \pm\tan\theta$	
55.ca	$\tan\theta, \cot\theta$			
56.ad_1				tan in D_1
57.X		$\tan^2$		
58.ba	$\tan^2$	0		
59.ad_2				$\tan^2$ in D_2
60.d_2c			$\tan^2$	
61.X		$\tan^4$		
62.bd_5		0		$\tan^4$ in D_5
63.ma	-0.4340			
64.X		$-.434\tan^2$		
65.bd4		0		$-.434\tan^2$ in D4
66.d_5c			$\tan^4$	
67.ma	0.1474			
68.X		$.1474\tan^4$		
69.d4a	$-.434\tan^2$			
70.+		sum		
71.ma	0.9965			
72.+	next inst 77	$\cos\theta$		
73.d_1a	K'			
74.+		K'		
75.Aca	L'			

Table IV. Continued

Inst.	No.in A Reg.	No.in B Reg.	No. in C.Reg.	No. in memory
76.div	Next inst 55	0	$K'/L' = \cot\theta$	
77.bd$_4$		0		cos or cos/cot in D4
78.wt				
79.ma	$K'-L'$			
80.ckS	if +, to 94 if -, to 81			
81.d$_4$c			cos/cot	
82.d$_1$a	cot			
83.X		cos θ		
84.bd$_4$		0		cos in D4
85.mc			Vdt	
86.d$_4$a	cos θ			
87.X		dx		
88.ba	dx	0		
89.ad$_2$				dx in D_2
90.+		dx		
91.d$_1$a	cot θ			
92.div		0	dy	
93.cd$_5$	next ins 103			dy in d_5.
94.mc			Vdt	
95.d$_4$a	cos θ			
96.X		dx		
97.ba	dx	0		
98.ad$_2$				dx in D_2
99.d$_1$c			tan θ	
100.X		dy		

Table IV. Continued

Inst.	No. in A Reg.	No. in B Reg.	No. in C Reg.	No. in memory
101. bd_5		0		dy in D_5
102. wt				
103. d_3a	X_p			
104. +		X_p		
105. d_6a	X			
106. -		$X_p - X = K$		
107. bd_4		0		K in D_4
108. d_2a	dx			
109. +		dx		
110. d_4a	K			
111. div		0	dx/K	
112. cd_1				dx/K in D_1
113. d_1a	dx/K			
114. ckS	if +, to 115 if -, to 116			
115. wt	next ins 119			
116. d_2a	dx			
117. -		-dx = dx*		
118. bd_2		0		dx* in D_2
119. d_8a	Y_p			
120. +		Y_p		
121. d_7a	Y			
122. -		$Y_p - Y = L$		
123. bd_4		0		L in D4
124. d_5a	dy			
125. +		dy		

Table IV. Continued

Inst.	No.in A Reg.	No.in B Reg.	No.in C Reg.	No.in Memory
126.d_4a	L			
127.div		0	dy/L	
128.cd_1				dy/L in D_1
129.d_1a	dy/L			
130.ckS	if +, to 131 if -, to 133			
131.d_5a	dy			
132.+	next ins135	dy		
133.d_5a	dy			
134.-		-dy=dy*		
135.ba	dy*	0		
136.ad_5				dy* in D_5
137.+		dy*		
138.d_7a	Y			
139.+		Y+dy*=Y_n+1.		
140.bd_5		0		Y_n+1 in D_5
141.d_2a	dx*			
142.+		dx*		
143.d_6a	X			
144.+		X+dx*=X_n+1		
145.ba	X_n+1	0		
146.d_5c			Y_n+1	
147.am				X_n+1 in output
148.ad_1				X_n+1 in D_1
149.cm				Y_n+1 in output
150.ma	K'+L' or $\frac{1}{4}$			

Table IV. Continued

Inst.	No. in A Reg.	No. in B Reg.	No. in C Reg.	No. in memory
151.$+$		$K' + L'$ or $\frac{1}{4}$		
152.ma	z			
153.-		$(K'+L')-z$		
154.ba	$(K'+L')-z$	0		
155.ckS	if $+$, to 156 if -, to 160			
156.wt				
157.wt				
158.wt				
159.wt	next inst 1			
160.wt				
161.ma	old ma instruction			
162.$+$		old ma instruction		
163.ma	1/16			
164.$+$		new ma instruction		
165.bd4		0		new ma inst in D4
166.ma	$2^{-4}+2^{-14}$			
167.$+$		$2^{-4}+2^{-14}$		
168.ma	count no.			
169.$+$		new count no.		
170.ba	new count no	0		
171.wt				
172.d4c			new ma instruction	
173.cm				new ma inst in memory.
174.am				new count no. in memory.
175.ckS	if $+$, to 176 if -, to 177			

Table IV. Continued

Inst.	No.in A Reg.	No.in B Reg.	No.in C Reg.	No.in memory
176.wt	next ins159 (recycles)			
177.ma	0			
178.wt				
179.am				0 in output 4
180.ma	new am inst			
181.wt				
182.am				new am inst in memory
183.$\neq$		new am inst		
184.ck0	stop computer			

Table IV. Continued

Inst.	No.in A Reg.	No.in B Reg.	No.in C Reg.	No.in memory
1.ba		0		
2.ma	old am instruction			
3.$\not$		old am instruction		
4.ma	1/16			
5.$\not$		new am instruction		
6.bd_4		0		new am inst in D_4
7.ma	X_u			
8.md_8				Y_u in D_8
9.$\not$		X_u		
10.ma	S.C.			
11.div		0	X	
12.dga	Y_u			
13.cd_2				X in D_2
14.$\not$		Y_u		
15.ma	S.C.			
16.div		0	Y	
17.d_2a	X			
18.d_4m				new am inst in memory
19.wt				
20.wt				

Table V. Program of Entry Instructions.

Inst.	No.in A Reg.	No.in B Reg.	No.in C Reg.	No.in memory
21.am				X in memory
22.cm				Y in memory
23.mc			0	
24.ma	$\frac{1}{2}$			
25.$\neq$		$\frac{1}{2}$		
26.cm				0 in output 4
27.ck0	Computer stops			
28.mc			ma instr	
29.ma	$1-2^{-7}$			
30.$\neq$		$1-2^{-7}$		
31.cm				ma inst in memory
32.mc			0	
33.am				full scale in output 4
34.cm				0 in count no.
35.ck0	Computer stops			

Table V. Continued

BIBLIOGRAPHY

1. Ridenour, The Role of the Computer, Scientific American, Vol. 187, No.3, pp.116-130 (Sept. 1952).

2. Standards on Electronic Computers: Definition of Terms, 1950, IRE Standard 50IRE8.S1, reprinted in Proc. of IRE, Vol.39, No.3, pp.271-277 (March 1951).

3. Nelson, A Digital Computer for Airborne Control Systems, Tran of IRE, Prof Group on Electronic Computers, Vol.PGEC-1, pp.2-5 (December 1952).

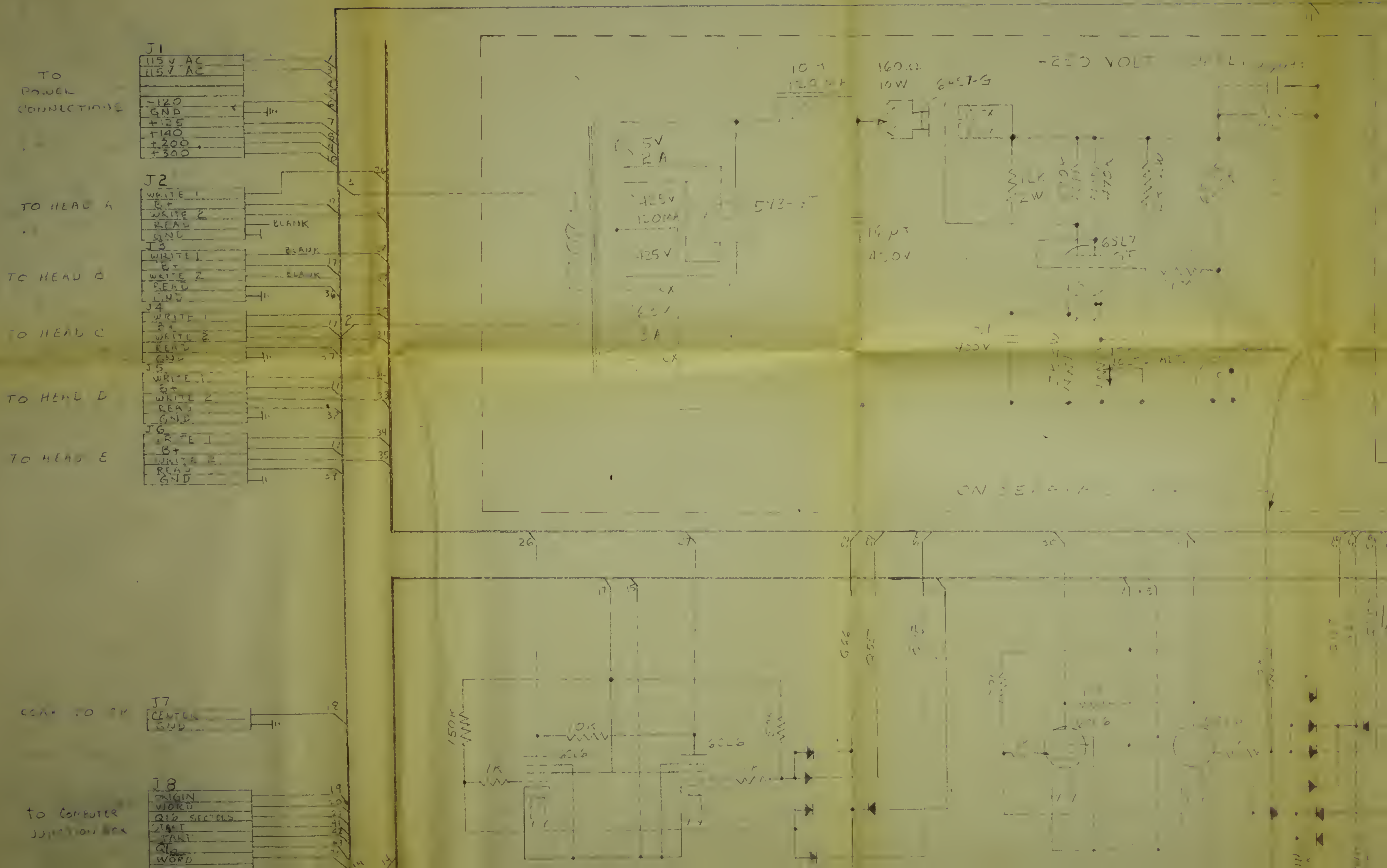
4. Bibliography of Coding Procedures, Math Tables and Other Aids to Comp, Vol.II, No.41, pp.47-48 (January 1953).

5. Hartree, Calculating Instruments and Machines, University of Illinois Press, Urbana, Illinois, 1949.

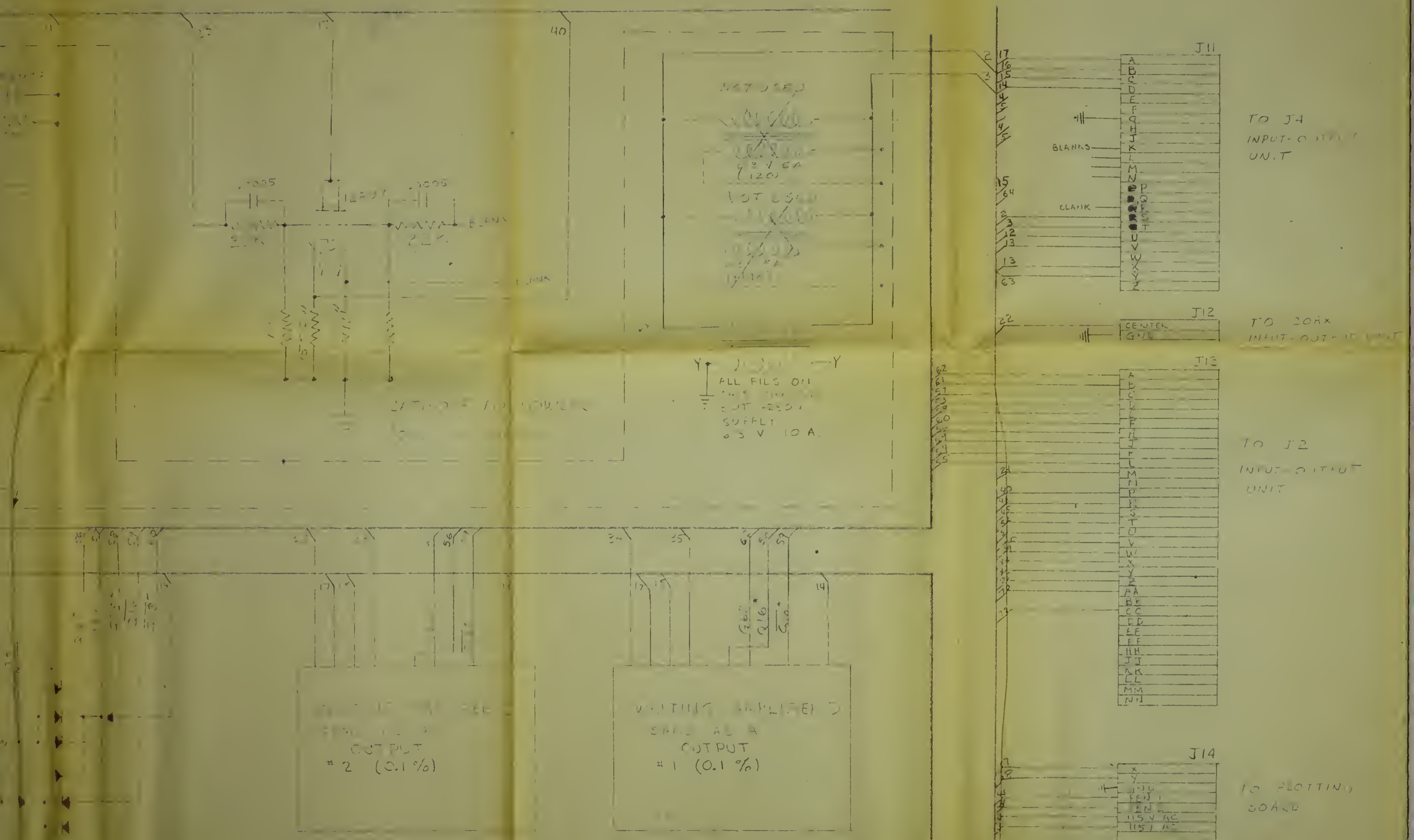
6. Rochester, Symbolic Programming, Trans of IRE, Prof Group on Electronic Computers, Vol.EC-2, No.1, pp.10-15 (March, 1953).

7. MacKnight and Adamson, Multichannel Analog Input-Output Conversion System for Digital Computer, Paper presented at IRE Convention, New York City, March, 1953, and to be reprinted in Proc. of IRE.

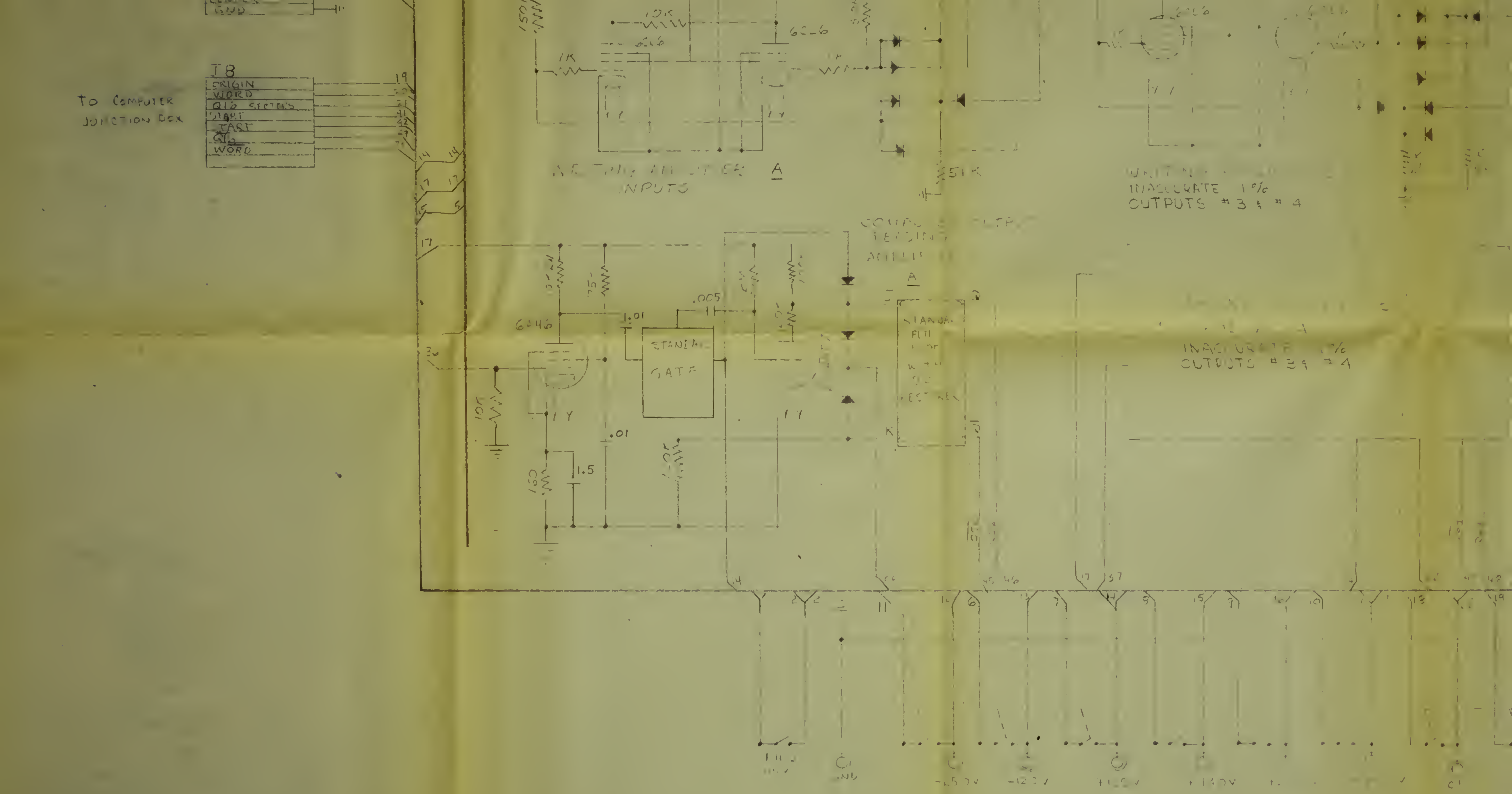
8. Background material on magnetic drum memory given in: ERA, Inc., Staff, High-Speed Computing Devices, pp.323-339, McGraw-Hill, New York, 1950



TO POWER CONNECTIONS
J1
115 V AC
115 V AC
-120
GND
+125
+140
+200
+300
J2
WRITE 1
B+
WRITE 2
READ
GND
BLANK
TO HEAD A
J3
WRITE 1
B+
WRITE 2
READ
GND
BLANK
TO HEAD B
J4
WRITE 1
B+
WRITE 2
READ
GND
TO HEAD C
J5
WRITE 1
B+
WRITE 2
READ
GND
TO HEAD D
J6
B+
READ
GND
TO HEAD E
J7
CENTER
GND
J8
ORIGIN
WORD
WORD
TO COMPUTER JUNCTION BOX
5V
2A
125V
120MA
125V
10W
6AS7-G
16 µF
6SL7
2W
150K
10K
1K



J11
TO J4
INPUT-OUTPUT
UNIT
BLANKS
J12
CENTER
GND
TO COAX
J13
TO J2
INPUT OUTPUT
UNIT
J14
TO PLOTTING
BOARD
115 V AC
NOT USED
ALL FILS ON
SUPPLY
6.3 V 10 A.
OUTPUT
2 (0.1 %)
OUTPUT
1 (0.1 %)


T8
ORIGIN
WORD
TO COMPUTER
JUNCTION BOX
WORD
INPUTS
51K
INACCURATE 1%
OUTPUTS #3 & #4
10K
1K
.005
.01
6046
STANDARD
GATE
1.5
INACCURATE 1%
OUTPUTS #3 & #4

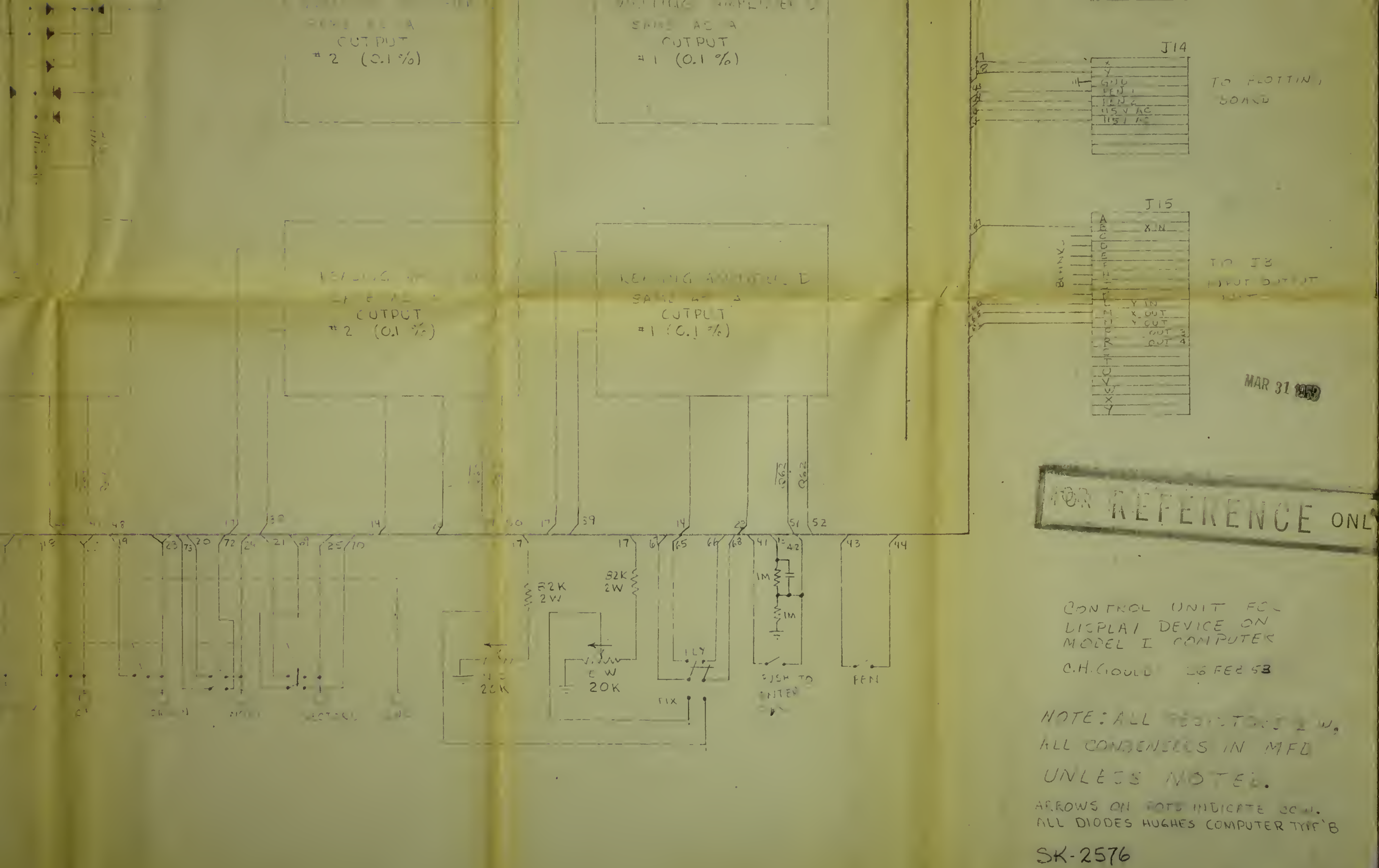


FIGURE I. SCHEMATIC OF CONTROL BOX

SECTOR	ORDER BAND 1	ORDER BAND 2	ORDER BAND 3	ORDER BAND 4	ORDER BAND 5	ORDER BAND 6	ORDER BAND 7	ORDER BAND 8 OR NUMBER BAND 1	NUMBER BAND 2	NUMBER BAND 3	NUMBER BAND 4	NUMBER BAND 5	NUMBER BAND 6	SECTOR
64		[illegible] 0,0,6 64F	d_2a 1,0,2 17E	ad_2 1,0,[illegible] [illegible]F	d_1a 0,0,1 15T									64
1	ba 0,0,1 1T	ba 1,0,8 1F	ba 0,0,3 1E	ba 0,0,6 145F										1
2	+ 5,0,1 16T	cm 1,1,4 31[illegible]	ckS [illegible],0,7 114F	am 10,[illegible],8 43F										2
3	md_5 1,[illegible],1 2T	d_4a 0,1,2 18[illegible]	ma 0,1,3 2E	bd_4 0,[illegible],5 84F	d_1c 5,0,6 99F	d_5c 1,0,6 140F								3
4	a_3m 6,3,7 36T	cd_6 1,0,5 17F	ca 0,0,[illegible] 55F	+ 0,0,3 104F	am 10,0,8 179F			am 1-16,2,2 21[illegible] mb 1-16,[illegible],4 2F						4
5	am 12,[illegible],8 174F	wt 31,[illegible],3 19[illegible]	+ 0,0,[illegible] 3[illegible]	mc 5,[illegible],6 32[illegible]	mc 1,[illegible],3 85F	+ [illegible],0,6 183F	+ 0,0,6 16[illegible]F				0.00000000000000			5
6	md [illegible],4,1 3T	ad_1 [illegible],0,5 56F	d_6a 7,0,7 105F	ca_1 [illegible],[illegible],7 1[illegible]8F		am 0,0,5 147F	wt 12,0,6 115F		[illegible]16					6
7		cm 15,[illegible],2 [illegible]	ma 0,[illegible],3 4E	mc 15,2,4 3[illegible]	d_3a 3,0,3 18F	ma 0,[illegible],5 153F	d_2a [illegible],0,5 116F		[illegible]8		0.00111111111111		OUTPUT 1 X_{n+1}	7
8	ma 2,3,8 41[illegible]	cm 15,[illegible],2	d_4a [illegible],0,7 86[illegible]	mc 15,[illegible],4	ad_1 0,[illegible],8 148F				[illegible]12		[illegible]$^{-4}$			8
9	ma 0,4,1 17T	cm 15,2,2	+ 0,0,3 5[illegible]	mc 15,[illegible],[illegible]	+ 0,0,6 164F	ckO 0,0,[illegible] 184F			[illegible]		1-[illegible]$^{-7}$			9
10	md_3 0,[illegible],1 4T	cm 15,[illegible],2	md_0 3,5,5 7F	mc 15,[illegible],4	[illegible] 0,0,5 57F	[illegible] 13,0,[illegible] 100F	d_1a 0,[illegible],3 1[illegible]9F	cm 0,6,5 149F	[illegible]14		- 7			10
11	- 3,[illegible],1 18T	cm 15,2,2	bd_4 0,0,3 6[illegible]	mc 15,[illegible],4	- 1,0,5 117F	bd_4 0,[illegible],0 165F			[illegible]6	COUNT #T	0.00000000000000		OUTPUT 2 Y_{n+1}	11
12	d_5c 0,0,1 5T	cm 15,[illegible],[illegible]	+ 0,0,3 14F	mc 15,[illegible],[illegible]	ma 0,3,7 150F	am [illegible],6,5 33[illegible]	wt 29,[illegible],1 37T	- 0,0,6 4[illegible]T	[illegible]10			INPUT 1		12
13		cm 15,[illegible],2	ma 0,5,3 7[illegible]	mc 15,[illegible],4	ca 1,6,3 [illegible]63	ca 0,0,6 32F	[illegible] 1,0,2 87F	ma 0,[illegible],8 166F		$\lvert X_p - X\rvert + \lvert Y_i - Y\rvert$				13
14	d_1a [illegible],0,1 6T	cm 15,[illegible],2	d_6a 7,[illegible],[illegible] [illegible]0F	mc 16,[illegible],[illegible]	od_2 3,0,6 118[illegible]	ckO 1,0,1 43T	+ 0,0,6 151F	d_1a [illegible],[illegible],7 44F	[illegible]15		$2^{-4}+2^{-14}$	INPUT 2 X		14
15		cm 15,[illegible],[illegible]	md_3 0,[illegible],[illegible] 8	mc 15,[illegible],[illegible]	[illegible] 0,[illegible],[illegible] 8F	ad_5 [illegible],0,[illegible] 33F	- 0,0,7 106F	+ 0,0,[illegible] 167F	[illegible]				OUTPUT 4 LIGHT	15
16	bd_1 [illegible],0,0 19[illegible]	cm 15,[illegible],[illegible]	ckO 0,[illegible],3 27[illegible]	mc 15,[illegible],[illegible]	cm 1,[illegible],1 34	ma 0,[illegible],5 15[illegible]F		ma 6,[illegible],5 180F	[illegible]11			INPUT 3 Y		16
17	wt 31,[illegible],1 4[illegible]	cm 15,[illegible],[illegible]	+ [illegible],[illegible],[illegible] [illegible]	mc 15,[illegible],[illegible]	+ [illegible],[illegible],5 [illegible]	bd_4 [illegible],[illegible],[illegible] [illegible]5F	od_4 [illegible],[illegible],7 [illegible]	ma [illegible],[illegible],7 168F	X_5		[illegible].00111111111111			17
18	X 0,0,1 7[illegible]	cm 15,[illegible],[illegible]	ckS [illegible],0,[illegible] 1[illegible]	mc 15,[illegible],[illegible]	- 0,0,3 159[illegible]	+ 0,[illegible],1 [illegible]	+ 0,[illegible],7 45F		[illegible]1[illegible]	COUNT [illegible] F		INPUT 4		18
19	ckO 0,[illegible],[illegible] 55 E	cm 15,[illegible],2	ma 0,[illegible],[illegible] 10	mc 15,[illegible],4	ma 0,[illegible],5 10F	d_1a 4,0,5 [illegible]19F	+ [illegible],[illegible],8 169F	ckO [illegible],0,6 175F	[illegible]5				OUTPUT 3	19
20	d_5a 0,0,1 [illegible]1F	cm 15,[illegible],2	[illegible] 1,[illegible],[illegible] 194F	mc 15,[illegible],[illegible]			d_5a 0,[illegible],3 [illegible]F		[illegible]		[illegible].00000001[illegible]0111	INPUT 5		20
21		cm 15,[illegible],[illegible]	div 0,[illegible],2 11[illegible]	mc 1[illegible],[illegible],4	div [illegible],[illegible],5 11F	d_5c 7,0,5 60F	[illegible] [illegible],[illegible],5 108F	ba 0,0,[illegible] 170F	[illegible]1					21
22	+ 0,0,1 [illegible]T	cm 15,2,[illegible]	d_1a [illegible],[illegible],3 34F	mc 15,[illegible],[illegible]	d_5a 6,0,6 1[illegible]1F							INPUT 6		22
23		- 0,0,3 21F	ckS [illegible],[illegible],0 155[illegible]	wt [illegible],[illegible],[illegible] 171F	d_5a 5,[illegible],7 1[illegible]3F	wt 78,0,[illegible] 170F			Y_{16}		1.00000111011110			23
24	bd_5 [illegible],0,8 [illegible]5T	mc 16,[illegible],[illegible] [illegible]	ma 10,[illegible],5 77F	cd_5 15,0,[illegible] 4F	wt 32,[illegible],4 181F	ma 15,4,0 177F			Y_8					24
25		mc 14,[illegible],2 [illegible]3[illegible]	bd_2 0,[illegible],3 [illegible]	cd_5 14,[illegible],[illegible]	+ 1,0,5 1[illegible]0F	wt 32,[illegible],4 156F			Y_{12}					25
26		mc 13,[illegible],2	+ 0,0,[illegible] 35F	cd_5 13,[illegible],[illegible]		wt [illegible]9,0,3 160F			Y_4			INPUT 7		26
27		mc 12,4,2	d_2c 6,0,3 23F	cd_5 12,[illegible],4	[illegible] 0,0,3 58F	+ 0,0,5 109F	wt 24,0,[illegible] 158F		Y_{14}					27
28		mc 11,[illegible],[illegible]	d_5a 0,0,1 36F	cd_5 11,[illegible],4	d_7a 2,0,6 121F				Y_6					28
29		mc 10,[illegible],2	ad [illegible],0,5 59F	$cd_5$10 ,[illegible],[illegible]	d_4a 6,0,7 110F				Y_{10}					29
30	+ 16,0,4 37F	mc 9,4,2	- 0,0,3 47F	cd_5 9,[illegible],4	ma 0,[illegible],5 67[illegible]	+ 1,0,5 152F	- 1,0,[illegible] 1[illegible]4F	d_1a [illegible],0,1 [illegible]4T	$Y_{[illegible]}$			INPUT 8		30
31		mc 8,[illegible],[illegible]	ba 0,[illegible],3 80F	cd_5 8,[illegible],4					Y_{15}		0.[illegible]0100101101111			31
32		mc 7,[illegible],2	ba 0,[illegible],3 48F	cd_5 7,[illegible],4	[illegible] 0,0,4 [illegible]F	- [illegible],[illegible],5 1[illegible]2[illegible]			Y_7					32
33		mc 6,[illegible],[illegible]	d 0,[illegible],6 [illegible]9F	cd_5 [illegible],[illegible],4	ba 0,0,7 135F				Y_{11}					33
34	+ [illegible]0,[illegible],1 [illegible]5T	mc 5,4,[illegible]	am 0,[illegible],[illegible] [illegible]	cd_5 [illegible],[illegible],[illegible]	bd_4 1,[illegible],8 143F				Y_3			INPUT 9		34
35	div 7,[illegible],1 [illegible]T	mc [illegible],[illegible],2	[illegible] 0,0,5 [illegible]	cd_5 4,[illegible],4	d_2c 6,0,4 60F	+ 0,0,6 [illegible]0F	ad_5 1,[illegible],[illegible] 1[illegible]F		Y_{13}		$X_p - X$ [illegible] $- \lvert Y_p - Y\rvert$				35

31		mc 3,[illegible],2	ba 0,0,3 83F	cd 0,[illegible],4					Y_{15}		0.0100101101111			31
32		mc 7,[illegible],2	ba [illegible],[illegible],3 45F	cd 7,[illegible],4	[illegible] 0,0,4 68F	- [illegible],[illegible],5 1[illegible]2F			Y_7					32
33		mc 6,[illegible],2	d 0,[illegible],6 89F	cd_8 0,0,4	ba 0,0,7 135F				Y_{11}					33
34	+ [illegible],[illegible],1 25T	mc 5,4,2	am 0,[illegible],3 49F	cd [illegible],[illegible],4	bd_4 1,0,8 123F				Y_3			INPUT 9		34
35	div 0,[illegible],1 6T	mc 4,[illegible],2	[illegible] [illegible],[illegible],5 [illegible]	cd 4,[illegible],4	d_2c 0,0,4 60F	+ 0,0,6 90F	ad_5 1,0,[illegible] 156F		Y_{13}	$\lvert Y_p - \rvert - \lvert Y_p - Y \rvert$				35
36	ma 0,4,1 16T	mc 3,4,2	ckS 1,[illegible],3 50F	cd [illegible],0,4	ckS 1,0,5 80F				Y_5				-------	36
37		mc [illegible],[illegible],2	st [illegible],[illegible],[illegible] [illegible]	cd [illegible],[illegible],4	ad_1 3,[illegible],3 25F	d_1a 3,0,4 91F	div 0,0,[illegible] 111F	d_5a 7,0,3 134F	Y_9		$1-2^{-14}$			37
38	+ 6,0,1 27[illegible]	mc 1,[illegible],2	d_8[illegible] 1,[illegible],3 12[illegible]	cd 1,0,4	cd_7 0,[illegible],2 12F	+ 0,0,6 137F			Y_1			INPUT 10 [illegible]	-------	38
39		mc 0,4,2	[illegible] 0,0,5 [illegible]1F	cd_8 0,0,4	mc [illegible],[illegible],6 4F									39
40		d_5a 5,0,[illegible] 13F	d_1a 0,0,0 73F	bd_5 4,0,2 101F	d_4c 3,0,4 31F	d_7a 0,0,3 138F					0.000000000000000		-------	40
41		ma 16,[illegible],[illegible] 24[illegible]	cd_2 0,[illegible],3 13[illegible]	ad_3 1,[illegible],[illegible] 5F	+ 0,0,[illegible] 52F	st 6,0,0 178F					ma 16,[illegible],4			41
42	d_3a 0,[illegible],5 [illegible]61	ma 16,[illegible],[illegible] [illegible]	d_6a 3,[illegible],[illegible] 26F	div 0,[illegible],4 92F	+ 1,0,[illegible] 74F						[illegible].0011111111110		-------	42
43		d a 0,[illegible],4 [illegible]	+ 0,[illegible],[illegible] 14[illegible]	x [illegible],[illegible],3 61F		d_4 [illegible],0,5 95F								43
44		st [illegible],[illegible],[illegible] 0F	+ [illegible],0,1 39[illegible]										-------	44
45		[illegible] [illegible],[illegible],4 75	ma [illegible],[illegible],[illegible] 1[illegible]	d_1[illegible] [illegible],[illegible],[illegible] 82F	[illegible] 0,0,4 96F									45
46	ba [illegible],0,1 [illegible]	st 15,[illegible],3 10[illegible]F	+ 0,[illegible],[illegible] 1[illegible]								0.000000010001110		-------	46
47		+ [illegible],0,[illegible] 1[illegible]	div 0,[illegible],[illegible] 1[illegible]	div [illegible],[illegible],3 76F								-------		47
48	ckS 0,0,1 29[illegible]	d_4[illegible] [illegible],[illegible],3 170[illegible]	+ [illegible],0,[illegible] 137[illegible]	ba [illegible],[illegible],5 3[illegible]F										48
49	st 20,[illegible],1 45[illegible]	ma [illegible],[illegible],[illegible] 15F	+ [illegible],[illegible],[illegible] 7[illegible]	d [illegible],[illegible],[illegible] [illegible]										49
50	st 16,[illegible],1 33[illegible]	[illegible] [illegible],0,4 3F	bd_5 [illegible],[illegible],4 1[illegible]	d c 2,[illegible],[illegible] 172F	ckS [illegible],[illegible],[illegible] [illegible]						0.0000001[illegible]01110		-------	50
51	[illegible] 0,[illegible],1 31T	div 0,[illegible],[illegible] 1[illegible]	d [illegible],[illegible],2 [illegible]	div [illegible],[illegible],[illegible] 54[illegible]								-------		51
52	+ 0,[illegible],1 [illegible]	st 1,[illegible],[illegible] 1[illegible]				st 1,[illegible],4 40[illegible]					2^{-14}			52
53	+ [illegible],[illegible],1 [illegible]	+ [illegible],[illegible],[illegible] 7[illegible]	div [illegible],[illegible],[illegible] 127[illegible]		cm 14,1,1 17[illegible]	m [illegible],4,[illegible] [illegible]1F								53
54	[illegible] [illegible],[illegible],1 1[illegible]	cd_1 [illegible],[illegible],[illegible] 11[illegible]		[illegible] [illegible],[illegible],[illegible] 141									-------	54
55	[illegible] [illegible],[illegible],1 [illegible]	[illegible] [illegible],[illegible],[illegible] 71[illegible]										-------		55
56	- [illegible],[illegible],1 1[illegible]	- [illegible],[illegible],[illegible] [illegible]	[illegible] 1[illegible],[illegible],[illegible] [illegible]								[illegible].11[illegible]1111[illegible]0110			56
57	[illegible]	+ [illegible]		[illegible]										57
58	[illegible]	[illegible]	[illegible]	[illegible]							[illegible]		-------	58
59		+ [illegible]	[illegible]	[illegible]	+ [illegible]	[illegible]					$1-2^{-7}$	-------		59
60	- [illegible],[illegible],1 [illegible]	+ [illegible]	[illegible]	+ [illegible]										60
61		[illegible]	[illegible]	[illegible]										61
62	[illegible]	[illegible]	[illegible]	[illegible]									-------	62
63	[illegible]	+ [illegible]									1.[illegible]1011[illegible]111			63

[illegible] 1 [illegible] used for [illegible].